UNDERSTANDING THE STATE MONAD

```
stackManip = do
                push (Just 3)
                pop
```

```
push (Just 3) >> pop
push (Just 3) >>= (\_ → pop )
State ( \xs → (Nothing, 3:xs) ) >>= (\_ → State (\(x:xs) → (x,xs))
State (\s → let (a , newState) = runState State ( \xs → (Nothing , (Just 3):xs) ) s
                in runState ((\_ → State ( \(x:xs) → (x,xs) ) a ) newState ) )
```

```
runState State (\s → let (a , newState) = runState State ( \xs → (Nothing , (Just 3):xs) ) s
                        in runState ((\_ → State ( \(x:xs) → (x,xs) ) a ) newState ) ) [(Just 1) , (Just 2) ]

runState State ([(Just 1) , (Just 2) ] → let ( a , newState) = runState State ( \xs → ( Nothing, (Just 3):xs) )  [(Just 1) , (Just 2) ]
                                          in runState ((\_ → State ( \(x:xs) → (x,xs) ) a ) newState ) )  [(Just 1) , (Just 2) ]

runState State ([(Just 1) , (Just 2) ] → let ( a , newState) = runState State (  [(Just 1) , (Just 2) ] → (Nothing, (Just 3):xs) )  [(Just 1) , (Just 2) ]
                                          in runState ((\_ → State ( \(x:xs) → (x,xs) ) a ) newState ) )  [(Just 1) , (Just 2) ]

runState State ([(Just 1) , (Just 2) ] → let ( , [(Just 3) , (Just 1) , (Just 2) ]) = runState State (  [(Just 1) , (Just 2) ] → (Nothing, (Just 3):xs) )  [(Just 1) , (Just 2) ]
                                          in runState ((\_ → State ( \(x:xs) → (x,xs) ) Nothing ) [(Just 3) , (Just 1) , (Just 2) ] ) )  [(Just 1) , (Just 2) ]

runState State ([(Just 1) , (Just 2) ] → let ( , [(Just 3) , (Just 1) , (Just 2) ]) = runState State (  [(Just 1) , (Just 2) ] → (Nothing, (Just 3):xs) )  [(Just 1) , (Just 2) ]
                                          in runState (State ( \(x:xs) → (x,xs) ) [(Just 3) , (Just 1) , (Just 2) ]  )  [(Just 1) , (Just 2) ]

runState State ([(Just 1) , (Just 2) ] → let ( , [(Just 3) , (Just 1) , (Just 2) ]) = runState State (  [(Just 1) , (Just 2) ] → (Nothing, (Just 3):xs) )  [(Just 1) , (Just 2) ]
                    in runState (State ( \ ((Just 3) : [ (Just 1) , (Just 2) ])→ ((Just 3),[(Just 1) ,(Just 2)]) [(Just 3) , (Just 1) , (Just 2) ]  )  [(Just 1) , (Just 2) ]

                                          ((Just 3),[(Just 1) ,(Just 2)])
```